

计算机视觉第一次作业

张博彦

October 2023

1 实验目标

实现对彩色图像的灰度处理，并使用高斯一阶差分滤波器计算图像梯度，进而执行非极大值抑制和阈值操作及连接，从而实现 canny 边缘检测的功能。

2 过程分析

- 首先根据公式 $\text{Gray} = 0.299 * R + 0.587 * G + 0.114 * B$ 将彩色图像转化为灰度图
- 对二维高斯分布进行 x 和 y 方向求偏导

$$\frac{\partial G}{\partial x} = \left(-\frac{1}{2\pi\sigma^4}\right) * xe^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

$$\frac{\partial G}{\partial y} = \left(-\frac{1}{2\pi\sigma^4}\right) * ye^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

- 根据偏导得到 x 方向核 y 方向的高斯卷积核
- 使用卷积核与灰度图进行卷积，并记录每个点的梯度方向
- 由于使用 $3*3$ 的卷积核，故只有 $0^\circ, 45^\circ, 90^\circ, 135^\circ$ 四个方向，所以将梯度方向中：
 - 22.5° 22.5° 和 157.5° 202.5° 方向的值设为 0
 - 22.5° 67.5° 和 202.5° 247.5° 方向的值设置为 45°
 - -90° -67.5° 、 67.5° 112.5° 和 247.5° 270° 方向的值设为 90°
 - -67.5° -22.5° 和 112.5° 157.5° 方向的值设为 135°

- 根据梯度的角度值遍历除图像边缘的所有点
 0° 为将该点与 $(-1, 0)$ 和 $(1, 0)$ 两点进行比较
 45° 为将该点与 $(1, 1)$ 和 $(-1, -1)$ 两点进行比较
 90° 为将该点与 $(0, -1)$ 和 $(0, 1)$ 两点进行比较
 135° 为将该点与 $(-1, 1)$ 和 $(1, -1)$ 两点进行比较
 如果该点值小于两个点中的任意一个，则判断为非边缘，将值设为 0
- 设置阈值的下限 lower 和上限 upper，大于 upper 的判定为边缘 255，低于 lower 的判定为非边缘 0
- 对于 lower upper 之间的值，以下四种情况： $(-1,0)$ 和 $(1,0)$ 处的点都为边缘 $(-1,1)$ 和 $(1,-1)$ 处的点都为边缘 $(0,-1)$ 和 $(0,1)$ 处的点都为边缘 $(1,-1)$ 和 $(-1,1)$ 处的点都为边缘如果该点满足任意其一，将该点判定为边缘 255，否则判定为非边缘 0
- 分别进行上述步骤后初步实现了 canny 边缘检测的功能

3 结果描述

最终实验结果如下图

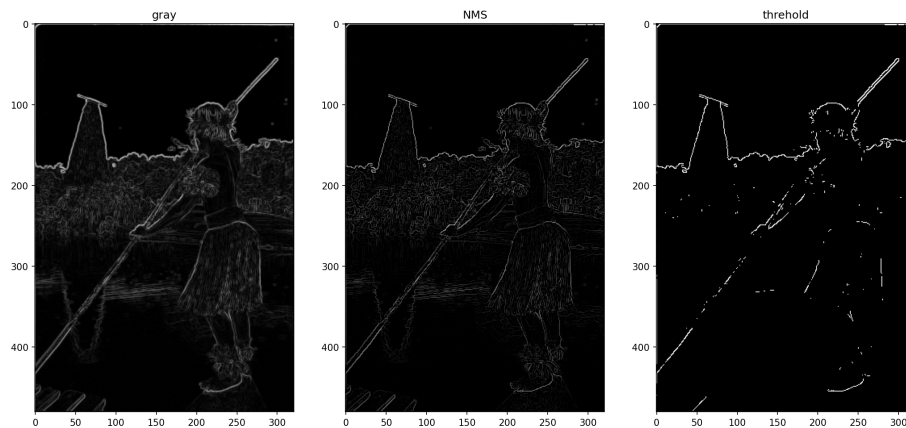


图 1: 实验结果

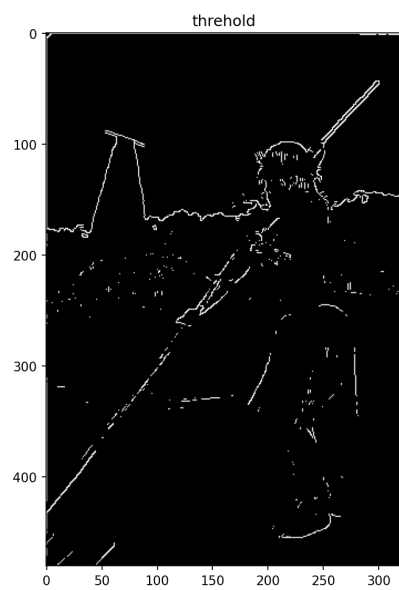


图 2: lower=10,upper=17

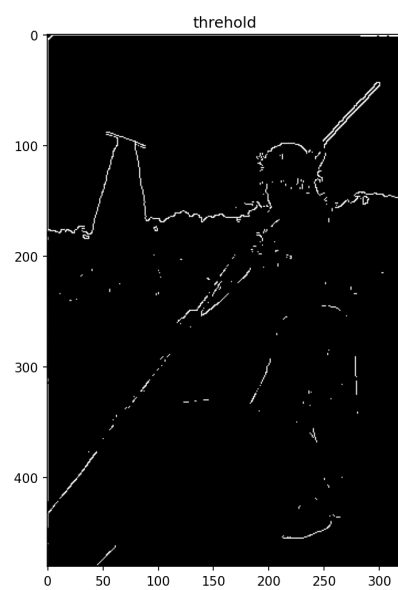


图 3: lower=10,upper=20

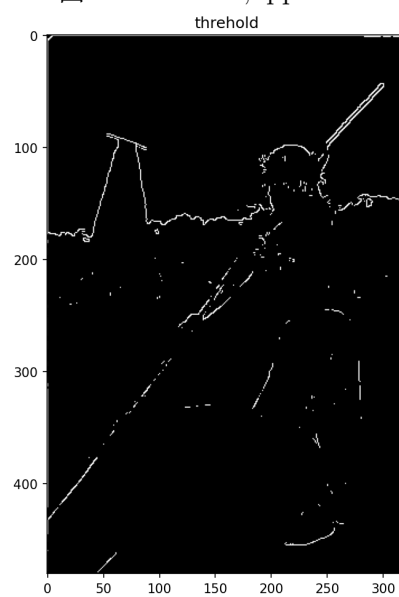


图 4: lower=14,upper=20

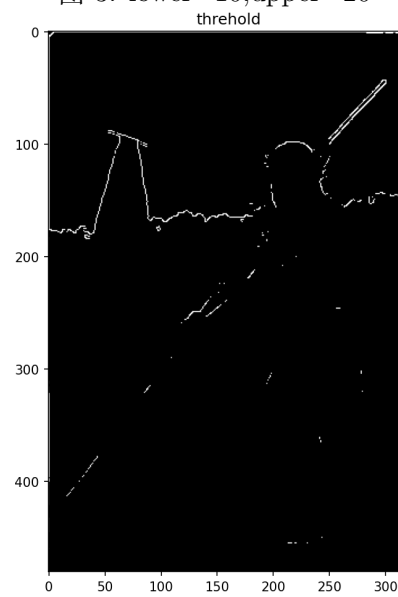


图 5: lower=20,upper=27

图 6: 改变 lower 和 upper 效果对比

当在实验中改变 lower 和 upper 时发现，当 lower 在 14 左右，upper 在 20 左右有较好的边缘检测效果，当 lower 和 upper 过低时会导致许多伪边缘被判定为真边缘，当 lower 和 upper 过高时会导致许多真边缘被判定为伪边缘。



图 7: 使用 cv2 库中的 canny 算法得到的效果

同时，使用 cv2 库中的 canny 算法与自己手动实现的进行对比，发现 canny 算法的效果会好很多，可能的原因是高斯一阶差分滤波器并不适合处理该张图片，使用 Sobel 算子进行处理可能有更好的检测效果

4 总结

通过该实验，具体了解到了 canny 边缘检测算法的实现过程，在实验过程中，最初对于梯度的方向 θ 的作用并不了解，所以直接使用周围 8 个点进行非极大值抑制，导致效果极差，然后经过网上查找了解到应该在梯度方向上判断是否为极大值。同时当进行阈值操作及连接时，刚开始只知道将大于 upper 的设为 255，小于 lower 的设为 0，但进行该操作后发现效果并不理想，很多非边缘被保留，最后从网上得知在进行非极大值抑制后图像可能存在强边缘，弱边缘和非边缘，上诉操作仅处理了强边缘和非边缘，对于弱边缘并未进行处理，因此对于该类图像点，需进行单独的判断，检测该点周围是否存在边缘点（值为 255），如果存在，则该点也为边缘，否则就认为为非边缘。最终，在经过方法的不断改进后，手动实现的 canny 边缘检测

虽然与 canny 算法仍有较大的差距、但也取得了较好的效果。